Classification using generative models
Classification problems
Problem: Create a program that, given an element from the input space $\mathcal{X}$, returns the element’s corresponding label from the output space $\mathcal{Y}$

**Classification problem**: $\mathcal{Y}$ is discrete (and typically finite) set
Examples:

- Spam filtering
  \[ \mathcal{X} = \{ \text{all emails} \}, \quad \mathcal{Y} = \{ \text{ham, spam} \} \]

- Intrusion detection
  \[ \mathcal{X} = \{ \text{all network traffic logs} \}, \quad \mathcal{Y} = \{ \text{benign, malicious} \} \]

- News analysis
  \[ \mathcal{X} = \{ \text{all news articles} \}, \quad \mathcal{Y} = \{ \text{politics, sports, business, . . .} \} \]
Model data as random variables

- **Feature vector**: random vector \( X = (X_1, X_2, \ldots, X_d) \)
- **Label**: a discrete random variable \( Y \)
- \( X \) and \( Y \) may be dependent!

**Typical goal**: create a classifier \( f : \mathcal{X} \to \mathcal{Y} \) with low **error rate**

\[
\text{err}[f] = \Pr(f(X) \neq Y)
\]
Iris dataset
Iris dataset (Fisher, 1936)

- 3 classes of iris plants $\mathcal{Y} = \{\text{Setosa, Versicolor, Virginica}\}$
- Take some measurements of each iris plant

- Training data: 40 examples from each class; test data: 10 examples per class
- Problem: Create a program that, given the measurements of an iris plant, returns the class that the plant belongs to
Generative models for classification
Generative model (for classification): a family of probability distributions for $(X,Y)$, each with the following form:

- Specify marginal distribution $p_Y$ of $Y$ (class prior)

- For each $k \in \mathcal{Y}$, specify conditional distribution of $X$ given $Y = k$ (class conditional distributions)

Example: Normal generative model
How to create classifier based on a distribution from the generative model?

- You have: $\hat{p}_Y$ and $\hat{p}_{X|Y=k}$ for each $k \in \mathcal{Y}$
- You want: $\hat{f}: \mathcal{X} \rightarrow \mathcal{Y}$
Generative model for iris dataset
Normal generative model for iris dataset using $x = \text{sepal length}$
Maximum likelihood estimation (MLE) of $\pi_k, \mu_k, \sigma_k^2$ for each $k \in Y$:

$$\hat{\pi}_k = \frac{\# \text{ training examples with label } k}{\# \text{ training examples}}$$

$$\hat{\mu}_k = \text{average value of } x \text{ among examples with label } k$$

$$\hat{\sigma}_k^2 = \text{average value of } (x - \hat{\mu}_k)^2 \text{ among examples with label } k$$

<table>
<thead>
<tr>
<th>$k$</th>
<th>setosa (1)</th>
<th>versicolor (2)</th>
<th>virginica (3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\hat{\pi}_k$</td>
<td>1/3</td>
<td>1/3</td>
<td>1/3</td>
</tr>
<tr>
<td>$\hat{\mu}_k$</td>
<td>4.99</td>
<td>5.93</td>
<td>6.61</td>
</tr>
<tr>
<td>$\hat{\sigma}_k$</td>
<td>0.31</td>
<td>0.47</td>
<td>0.68</td>
</tr>
</tbody>
</table>
def learn(train_x, train_y, num_classes=3):
    return [(np.mean(train_y == k), np.mean(train_x[train_y == k]),
             np.var(train_x[train_y == k])) for k in range(num_classes)]

def predict(params, test_x):
    log_posterior = np.array([np.log(prior) - np.log(sigma2) / 2 -
                              (test_x - mu) ** 2 / (2*sigma2) for prior, mu, sigma2 in
                              params])
    return np.argmax(log_posterior, axis=0)
Resulting classifier:

\[ \hat{f}(x) = \arg \max_{k \in Y} \hat{\pi}_k \cdot \frac{1}{\sqrt{2\pi \hat{\sigma}_k^2}} \exp \left( -\frac{(x - \hat{\mu}_k)^2}{2\hat{\sigma}_k^2} \right) \]

Training error rate of \( \hat{f} \): 24%
Test error rate of \( \hat{f} \): 40%
Bivariate normal distributions
Now use two features: \( x = (x_1, x_2) = (\text{sepal length}, \text{petal length}) \)

Need generative model with class conditional distributions suitable for two-dimensional feature vectors
Bivariate normal distribution: 5 parameters (up from 2)

Resulting classifier has test error rate 10% (down from 40%)
If $Z_1$ and $Z_2$ are independent random variables and each is a standard normal random variable, then distribution of $Z = (Z_1, Z_2)$ is **standard bivariate normal**

$$p(z_1, z_2) = \frac{1}{2\pi} \exp\left(-\frac{z_1^2 + z_2^2}{2}\right) = \phi(z_1)\phi(z_2)$$
What is distribution of $X = aZ_1 + bZ_2$?
What is distribution of \((X_1, X_2)\), with \(X_1 = aZ_1 + bZ_2\) and \(X_2 = cZ_1 + dZ_2\)?
Density function for \((X_1, X_2)\) where \(X_1 = Z_1 + \frac{1}{3}Z_2\) and \(X_2 = \frac{1}{3}Z_1 + Z_2\):
General bivariate normal distribution:

\[
p(x_1, x_2)(x_1, x_2) = \frac{1}{2\pi \sqrt{\det(\Sigma)}} \exp \left( -\frac{1}{2} [x_1 - \mu_1]^T \Sigma^{-1} [x_1 - \mu_1] \right)
\]
Fitting bivariate normal distribution to data using MLE:

\[ \hat{\mu}_1 = \text{average value of } x_1 \text{ in dataset} \]
\[ \hat{\mu}_2 = \text{average value of } x_2 \text{ in dataset} \]
\[ \hat{\Sigma}_{1,1} = \text{average value of } (x_1 - \hat{\mu}_1)^2 \text{ in dataset} \]
\[ \hat{\Sigma}_{1,2} = \hat{\Sigma}_{2,1} = \text{average value of } (x_1 - \hat{\mu}_1)(x_2 - \hat{\mu}_2) \text{ in dataset} \]
\[ \hat{\Sigma}_{2,2} = \text{average value of } (x_2 - \hat{\mu}_2)^2 \text{ in dataset} \]

(In context of generative models, do this for each class)
Multivariate normal distributions
General multivariate normal distribution in $d$-dimensions:

$$p_X(x) = \frac{1}{\sqrt{(2\pi)^d \det(\Sigma)}} \exp\left(-\frac{1}{2}(x - \mu)^T \Sigma^{-1} (x - \mu)\right)$$

**MLE:**

$$\hat{\mu}_i = \text{average value of } x_i \text{ in dataset}$$

$$\hat{\Sigma}_{i,j} = \text{average value of } (x_i - \hat{\mu}_i)(x_j - \hat{\mu}_j) \text{ in dataset}$$